

HOMEWORK #8:

Software Upgrade Upgrade.

Due Monday, **December the 2nd**, 11:59pm

For this assignment, you need to submit a file called 'maxheap.h' and any other necessary implementation files.

Remember to put your name and section at the top of all your files.

Problem

Your "Binary Search Tree" implementation has been a hit for professor Farnsworth and he has decided to licence your code to other robots and other professors throughout the galaxy. However, he has received several requests from potential clients that would like a different kind of tree. Instead of the elements of the tree following the Search order, they are interested in a "best" element to be at the root of the tree. So that when they look at the root, they always see the "best" element, and if they remove the root, the next "best" element takes its place. Your extensive computer science experience tells you that what he now wants is a **max-heap**!



I have more work for you.

Your job is to write a max-heap implementation!

Testing

Use the provided tester file to check if your implementation is working correctly.

- The program 'maxheapester.cpp' uses the 'maxHeap' class and the intended output is 'maxheapoutput.txt'.

Notice:

1. You are expected to derive the 'AbstractHeap' class.
2. You are also expected to implement the "Big-3" for your 'maxHeap' class
3. The function 'top()' is expected to "throw" errors.

Useful Hints

1. Carefully read the comments of each member function.
2. Write down an algorithm for the function before you start coding it.
3. Develop your member functions one at a time, starting from the simplest ones.
Move to the next function only after the previous one has been tested.
Trying to code the whole class and then remove the bugs may prove to be too big a task.
4. Use the test functions one at a time.

More Hints

- Use a dynamically allocated array for storing the elements in the Heap.
- The following is *pseudo-code* for "Pretty Printing" a heap stored in an 'm_data' container array with 'm_size' elements in it.:

```
void prettyPrint (int ix, int pad)
{
    string s(pad, ' ');
    if (ix > m_size)
        cout << endl;
    else{
        prettyPrint( right(ix), pad+4);
        cout << s << m_data[ix] << endl;
        prettyPrint( left(ix), pad+4);
    }
}
```

END.